

类文件结构

2022年1月13日 12:47

字节码文件的 平台、语言无关性.

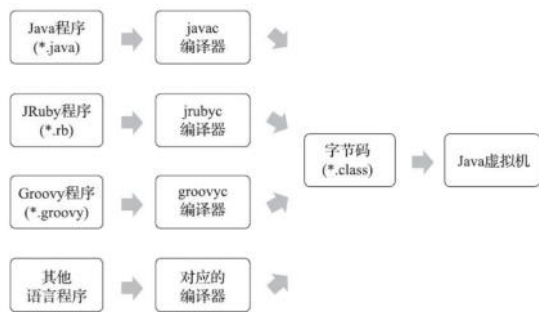


图6-1 Java虚拟机提供的语言无关性

Class文件的结构

① 定义

class文件: 一组以8个字节为基础单位的二进制流.

class文件结构中包含两种类型: { 无符号数, 表. 以 _info 结尾.

集合表示方法: 一个前置容量计数器 + 若干连续数据项.

② 魔数与class文件的版本

(1) 魔数: 每个class文件头4个字节.

作用: 判断是否为class文件.

(2) 次版本号: 第5、6字节.

主版本号: 第7、8字节.

JVM可以兼容以前版本.

③ 常量池: { 字面量 (Literal), 符号引用 (Symbolic References). 可以当作Java中的常量.

↓ 编译原理概念.

↓ 分类:

Javap - verbose 分析 class.

· 被模块导出或者开放的包 (Package)

· 类和接口的全限定名 (Fully Qualified Name)

· 字段的名称和描述符 (Descriptor)

· 方法的名称和描述符

· 方法句柄和方法类型 (Method Handle, Method Type, Invoke Dynamic)

· 动态调用点和动态常量 (Dynamically-Computed Call Site, Dynamically-Computed Constant)

④ 访问标识.

⑤ 类索引、父类索引与接口索引集合.



指向 Constant - class - info 的类描述符常量.



指向 Constant - utf8 - info 类型中的全限定字符串.

⑥ 字段表集合.

↓ 若干格式、接口或类中声明的变量

⑥ 字节表集合.

用于描述接口或类中声明的变量.

⑦ 方法表集合.

⑧ 属性表集合.

(1) code Java代码编译成的字节码指令.

字节码指令简介.

① 指令集架构 优先突出.

设计 操作码 一个字节

→ tableswitch、lookupswitch 除外.

放弃编译后代码的操作数对齐. → 省略填充和分隔符.

(1) 优. 小数据量, 高传输效率.

(2) 劣. 超过1字节的操作数合并 → 性能↓.

(3) 解释器执行模型.

```
do {  
    自动计算PC寄存器的值加1;  
    根据PC寄存器指示的位置, 从字节码流中取出操作码;  
    if (字节码存在操作数) 从字节码流中取出操作数;  
    执行操作码所定义的操作;  
} while (字节码流长度 > 0);
```

② 字节码与数据类型.

iload → int 类型.

a: reference

fload → float 类型.

指令 相完全独立. 每个指令 并非对应每个数据类型.

③ 加载和存储指令.

→ iload { 0, 1, 2 }

· 将一个局部变量加载到操作栈: iload、iload_<n>、lload、lload_<n>、fload、fload_<n>、dload、dload_<n>、aload、aload_<n>

· 将一个数值从操作数栈存储到局部变量表: istore、istore_<n>、lstore、lstore_<n>、fstore、fstore_<n>、dstore、dstore_<n>、astore、astore_<n>

· 将一个常量加载到操作数栈: bipush、sipush、ldc、ldc_w、ldc2_w、aconst_null、iconst_m1、iconst_<i>、lconst_<l>、fconst_<f>、dconst_<d>

· 扩充局部变量表的访问索引的指令: wide

④ 运算指令: 对两个操作数栈上的值进行计算.

- 加法指令: iadd、ladd、fadd、dadd
- 减法指令: isub、lsub、fsub、dsub
- 乘法指令: imul、lmul、fmul、dmul
- 除法指令: idiv、ldiv、fdiv、ddiv
- 求余指令: irem、lrem、frem、drem
- 取反指令: ineg、lneg、fneg、dneg
- 位移指令: ishl、ishr、iushr、lshl、lshr、lushr
- 按位或指令: ior、lor
- 按位与指令: iand、land
- 按位异或指令: ixor、lxor
- 局部变量自增指令: iinc
- 比较指令: dcmpl、dcmpl、fcmpl、fcmpl、lcmp